

This is a fairly elegant procedure if a number of calls to low level subroutines is required. (The coding suggested is dreadful - use a FOR-NEXT loop to load data.)

Background. to method.

The `USR()` command is capable of passing an argument to the subroutine being called. Usually a dummy (0) is passed. The argument is stored in 31009 & 31010 (^{LSB} 7921 H & ^{MSB} 7922 H). This method passes the start address of the required routine via this "Jump" routine. The RETURN in the subroutine called goes back to BASIC.

Conventional method.

13392 \equiv 3450 H st. add. Rom generate beep
13404 \equiv 345CH st. add. Rom generate sound.
(These are two subs. in Rom used as examples.)
line 10 pokes 3450 H as jump address.
line 30 pokes 345CH as jump address.
both initiated by `USR` command in lines 20 and 40. Note dummy argument passed.

MACHINE LANGUAGE CALLS

This simple VZ200/300 routine can save programmers from using lots of POKE commands in a Basic program when calling a lot of machine code subroutines.

Conventional method:

To call the address 13392 & 13404

10 POKE 30862,80:POKE 30863,52

20 `x=USR(0)`

30 POKE 30862,92:POKE 30863,52

40 `x=USR(0)`

New method:

10 `x=USR(13392):`

`x=USR(13404)`

Main program:

0 POKE 52992-65536,58:

POKE 52993-65536,33

1 POKE 52994-65536,

121:POKE 52995-

65536,50

2 POKE 52996-65536,13:

POKE 52997-65536,207

3 POKE 52998-65536,58:

POKE 52999-65536,34

4 POKE 53000-65536,121

:POKE 53001-65536,50

5 POKE 53002-65536,14:

POKE 53003-65536,207

6 POKE 53004-65536,195

:POKE 30862,0

7 POKE 30863,207

Feb 85 7(2)

New method.

52992-65536 = -12544 \equiv CFDD H } 15 bytes
53004-65536 = -12532 = CFDC H

(note that next two bytes are used also)
15 bytes in all.

Disassembled listing.

CFDD	3A 21 79	LD A, (7921 H)
B3	32 DD CF	LD (CFDDH), A
B6	3A 22 79	LD A, (7922 H)
B9	32 DE CF	LD (CFDEH), A
CFDC	C3 nn nn	JP nnnn
	(LSB)(MSB)	

Thus the argument passed by the `USR` command is read from 7921/2 H and written into CFDD/E H. which is then jumped to.

This simplifies the main line program significantly.